TITLE:
A proximal memoryless symmetric rank one method for minimizing composite functions (New Trends of Numerical Optimization in Advanced Information-Oriented Society)

AUTHOR(S):
成島, 康史; 中山, 舜民

CITATION:
成島, 康史 ...[et al]. A proximal memoryless symmetric rank one method for minimizing composite functions (New Trends of Numerical Optimization in Advanced Information-Oriented Society). 数理解析研究所講究録 2019, 2108: 196-205

ISSUE DATE:
2019-04

URL:
http://hdl.handle.net/2433/251934

RIGHT:

京都大学学術情報リポジトリ
KURENAI 紅
Kyoto University Research Information Repository

数理解析研究所講究録 第2108巻 2019年 196-205

A proximal memoryless symmetric rank one method for minimizing composite functions

横浜国立大学　成島康史 (Yasushi NARUSHIMA)
Yokohama National University
東京理科大学　中山舜民 (Shummin NAKAYAMA)
Tokyo University of Science

1　はじめに

この論文では，下記の無制約最適化問題を考える:

$$\min_{x\in\mathbb{R}^n} \quad f(x) := g(x) + h(x). \tag{1}$$

ただし，$g:\mathbb{R}^n\to\mathbb{R}$ は十分滑らかな凸関数で，$h:\mathbb{R}^n\to\mathbb{R}$ は必ずしも微分可能とは限らない凸関数であるとする．このような問題は，機械学習などで発生する問題で，近年注目を集めている．例えば，二乗損失を用いた ℓ_1 正則化学習を回帰に用いた手法は Lsso (Least absolute shrinkage and selection operator) と呼ばれ，

$$\min_{x\in\mathbb{R}^n} \quad \|Ax-b\|^2 + \lambda\|x\|_1$$

によってパラメータベクトル x を推定する．ここで，$\|\cdot\|$ は ℓ_2 ノルムを表し，$\|\cdot\|_1$ は ℓ_1 ノルムを表している．さらに，$A\in\mathbb{R}^{m\times n}$, $b\in\mathbb{R}^m$ は入力データであり，m は入力データの数を表す．また，$\lambda>0$ は正則化項を調整するパラメータである．このとき，この問題は (1) において，$g(x)=\|Ax-b\|^2$，$h(x)=\lambda\|x\|_1$ としたものに相当する．

問題 (1) に対しては近接勾配法 (proximal gradient method) と呼ばれる方法が有効であることが知られている．近接勾配法は反復法の一種で，任意の初期点 $x_0\in\mathbb{R}^n$ からスタートし，

$$x_{k+1} = \operatorname*{argmin}_{x\in\mathbb{R}^n}\left\{g(x_k) + \nabla g(x_k)^T(x-x_k) + h(x) + \frac{1}{2t_k}\|x-x_k\|^2\right\} \tag{2}$$

によって点列 $\{x_k\}$ を更新する．ここで，t_k は正のパラメータである．近接勾配法では，各反復で目的関数 $f(x)$ を x_k のまわりで 1 次近似した関数 $g(x_k)+\nabla g(x_k)^T(x-x_k)+h(x)$ を最小化することで点列を更新していることとなる．ただし，$\frac{1}{2t_k}\|x-x_k\|^2$ の項は，1 次近似が信頼できる領域から飛び出ないようにするペナルティ項である．ここで，簡単な計算から

$$\begin{aligned}&\operatorname*{argmin}_{x\in\mathbb{R}^n}\left\{g(x_k) + \nabla g(x_k)^T(x-x_k) + h(x) + \frac{1}{2t_k}\|x-x_k\|^2\right\}\\ &\qquad\qquad = \operatorname*{argmin}_{x\in R^n}\left\{h(x) + \frac{1}{2t_k}\|x-(x_k - t_k\nabla g(x_k))\|^2\right\}\end{aligned}$$

と書き直すことができるため，近接写像 (proximal mapping) を

$$\mathrm{Prox}_h(z) \equiv \operatorname*{argmin}_{x\in\mathbb{R}^n}\left\{h(x)+\frac{1}{2}\|x-z\|^2\right\} \tag{3}$$

によって定めれば，近接勾配法の反復式は

$$x_{k+1}=\mathrm{Prox}_{t_kh}\left(x_k-t_k\nabla g(x_k)\right)$$

によって再定義することができる．上記の式から，近接勾配法は t_k をステップ幅とした最急降下法に近接写像を施した方法であることがわかる．また，近接写像は関数 h が上述した ℓ_1 ノルムなどの場合には明示的に表現することができ，近接写像の計算時に (3) の最小化問題を解く必要がないことを注意しておく．

近接勾配法が最急降下法に基づいた方法であることを述べたが，一般的に無制約最適化問題に対する最急降下法は効果的ではないことが知られている．そこで，近接勾配法の改良を目的として，準ニュートン法に基づいた近接勾配法 (以下では，準ニュートン型近接勾配法と呼ぶ) が提案されている (例えば，[2,4,6,9] など)．それらの方法では，毎回の反復で目的関数の 2 次近似を最小化する．つまり，

$$x_{k+1}=\operatorname*{argmin}_{x\in\mathbb{R}^n}\left\{g(x_k)+\nabla g(x_k)^T(x-x_k)+\frac{1}{2}(x-x_k)^TB_k(x-x_k)+h(x)\right\} \tag{4}$$

によって点列 $\{x_k\}$ を更新する．ここで，B_k は $\nabla^2 g(x_k)$ の正定値対称な近似行列である．ここで，正定値対称な行列 A による重み付きノルムを

$$\|x\|_A=\sqrt{x^TAx}$$

によって定義すると，$(x-x_k)^TB_k(x-x_k)=\|x-x_k\|_{B_k}^2$ となることから，(4) は (2) において，ペナルティ項を重み付きノルムで置き換えたものに相当することがわかる．ここで，$H_k=B_k^{-1}$ とすると，近接勾配法のときと同様に，関係式：

$$\begin{aligned}&\operatorname*{argmin}_{x\in\mathbb{R}^n}\left\{h(x)+g(x_k)+\nabla g(x_k)^T(x-x_k)+\frac{1}{2}\|x-x_k\|_{B_k}^2\right\}\\&\qquad=\operatorname*{argmin}_{x\in\mathbb{R}^n}\left\{h(x)+\frac{1}{2}\left\|x-(x_k-H_k\nabla g(x_k))\right\|_{B_k}^2\right\}\end{aligned}$$

を用いて，準ニュートン型近接勾配法の反復式は

$$x_{k+1}=\mathrm{Prox}_h^{B_k}\left(x_k-H_k\nabla g(x_k)\right) \tag{5}$$

によって定義される．ただし，

$$\mathrm{Prox}_h^A(z)=\operatorname*{argmin}_{x\in\mathbb{R}^n}\left\{h(x)+\frac{1}{2}\left\|x-z\right\|_A^2\right\} \tag{6}$$

を正定値対称行列 A による重み付き近接写像とする．更新式 (5) から分かるように，準ニュートン型近接勾配法は，準ニュートン法に重み付き近接写像を施した方法であり，通常の近接勾配法よりも最適解の近傍における収束速度が速いことが期待できる．しかしながら，通常の近接写像とは異なり，$h(x)=\|x\|_1$ などの場合であっても，重み付き近接写像の値は直接計算することはできず，最適化問題 (6) を反復法などで近似的に解く必要

がある．ここで，(6) 自体も元の問題と同じく，微分可能な関数と微分不可能な関数の和の形で表されていることを注意しておく．したがって，外部反復 (5) が通常の近接勾配法よりも高速であったとしても，(6) を解くための内部反復に手間がかかってしまう恐れがある．そこで，Becker and Fadiliy [2] は行列 A が対角行列とランク 1 行列の和，つまり，$A = D + uu^T$ で，D が正の対角成分を持つ対角行列のときに，(6) を直接計算する方法を与え，それに基づく準ニュートン型近接勾配法を提案している．しかしながら，彼らの方法では近似行列 B_k の正定値性は保証されておらず，収束性も議論されていない．したがって，今回我々は，メモリーレス準ニュートン法の枠組みを用いて，重み付き近接写像が直接計算可能で，かつ，近似行列 B_k の正定値性を保証するようなメモリーレス準ニュートン型近接勾配法を提案し，その大域的収束性を議論する．

2　提案手法

まず最初に，今回の提案手法にとって重要な定理を紹介しておく．

定理 1. (Becker and Fadiliy [2])　$D \in \mathbb{R}^{n\times n}$ を対角成分がすべて正の対角行列とし，$u \in \mathbb{R}^n$ を任意のベクトルとする．行列 $A \in \mathbb{R}^{n\times n}$ が $A = D \pm uu^T$ で，さらに正定値であるとき，

$$\mathrm{Prox}_h^A(z) = D^{-1/2} \circ \mathrm{Prox}_{h\circ D^{-1/2}}(D^{1/2}z \mp v)$$

が成立する．ただし，$v = \alpha D^{-1/2}u$ であり，α は方程式

$$\left\langle u, z - D^{-1/2} \circ \mathrm{Prox}_{h\circ D^{-1/2}} \circ D^{1/2}(z \mp \alpha D^{-1}u)\right\rangle + \alpha = 0$$

の唯一解である．　□

上記の定理で，対角行列 D を単位行列 I，またはその定数倍を用いれば，重み付き近接写像 Prox_h^A は通常の近接写像 Prox_h を用いて計算可能である．したがって，通常の近接写像が陽に計算できる場合は，重み付き近接写像も陽に計算が可能である．

次に，近似行列 B_k の選択法について考える．通常の無制約最適化問題に対する準ニュートン法では，近似行列 B_k が $B_k \approx \nabla^2 g(x_k)$ であることが望まれる．ここで，$\nabla g(x_{k-1})$ の 1 次近似を考えると

$$\nabla g(x_{k-1}) \approx \nabla g(x_k) - \nabla^2 g(x_k)(x_k - x_{k-1}) \tag{7}$$

という関係式が得られる．よって，近似行列が満たすべき条件として

$$B_k s_{k-1} = y_{k-1}$$

を考えることができる．これをセカント条件と呼ぶ．ただし，

$$s_{k-1} = x_k - x_{k-1}, \quad y_{k-1} = \nabla g(x_k) - \nabla g(x_{k-1})$$

とする．セカント条件を満たす近似行列の更新公式として，DFP 公式，BFGS 公式，対称ランク 1 (Symmetric Rank 1, SR1) 公式などがよく知られているが，今回は定理 1 を利

用するために，SR1 公式に着目する．SR1 公式はその名の通り，ランク 1 の修正により近似行列を更新する方法で，その更新公式は

$$B_k = B_{k-1} + \frac{(y_{k-1} - B_{k-1}s_{k-1})(y_{k-1} - B_{k-1}s_{k-1})^T}{s_{k-1}^T(y_{k-1} - B_{k-1}s_{k-1})} \tag{8}$$

で与えられ，逆行列版の更新公式は

$$H_k = H_{k-1} + \frac{(s_{k-1} - H_{k-1}y_{k-1})(s_{k-1} - H_{k-1}y_{k-1})^T}{y_{k-1}^T(s_{k-1} - H_{k-1}y_{k-1})} \tag{9}$$

で与えられる．メモリーレス準ニュートン法では，準ニュートン法の更新公式においてひとつ前の近似行列 B_{k-1} の代わりに単位行列 I，もしくはスケーリングパラメータ $\gamma_{k-1} > 0$ を乗じた対角行列 $\gamma_{k-1}I$ で置き換えることで近似行列を作成する．例えば，更新式 (8) において，B_{k-1} を $\gamma_{k-1}I$ で置き換えると，近似行列として，

$$B_k = \gamma_{k-1}I + \frac{(y_{k-1} - \gamma_{k-1}s_{k-1})(y_{k-1} - \gamma_{k-1}s_{k-1})^T}{s_{k-1}^T(y_{k-1} - \gamma_{k-1}s_{k-1})}$$

が得られる．しかしながら，(元々の SR1 公式がそうであるように) 上記によって生成された行列は正定値であるとは限らないため，常に定理 1 を適用できるとは限らない．さらに，大域的な収束性を議論する場合には近似行列の有界性なども必要となる．そのため，今回我々は Cheng and Li [3] のスペクトラルスケーリングセカント条件 (以下では，SS セカント条件と呼ぶ) と Li and Fukushima [5] の修正セカント条件を組み合わせたセカント条件を考え，それに基づいたメモリーレス SR1 公式を提案する．

Cheng and Li は (7) の両辺にスケーリングパラメータ $\gamma_{k-1} > 0$ を乗じた近似式：

$$\gamma_{k-1}\nabla^2 g(x_k)(x_k - x_{k-1}) \approx \gamma_{k-1}(\nabla g(x_k) - \nabla g(x_{k-1})) \tag{10}$$

を考え，$B_k \approx \gamma_{k-1}\nabla^2 g(x_k)$ として，SS セカント条件：$B_k s_{k-1} = \gamma_{k-1}y_{k-1}$ を提案している．SS セカント条件における近似行列 B_k は $\nabla^2 g(x_k)$ の近似ではなく，$\gamma_{k-1}\nabla^2 g(x_k)$ であることから，スケーリングパラメータ γ_{k-1} をうまく選ぶことで，近似行列の条件数を抑え，数値的な安定性の向上が期待できる．今回我々は，Li and Fukushima に倣い，SS セカント条件の近似式 (10) に正則化項を加えた関係式：

$$\gamma_{k-1}(\nabla^2 g(x_k) + \nu_{k-1}I)(x_k - x_{k-1}) \approx \gamma_{k-1}(\nabla g(x_k) - \nabla g(x_{k-1}) + \nu_{k-1}(x_k - x_{k-1}))$$

を考える．ただし，ν_{k-1} は非負のパラメータとする．このとき，$B_k \approx \gamma_{k-1}(\nabla^2 g(x_k)+\nu_{k-1}I)$ とすると，修正 SS セカント条件

$$B_k s_{k-1} = \gamma_{k-1}z_{k-1}$$

が得られる．ただし，

$$z_{k-1} = y_{k-1} + \nu_{k-1}s_{k-1}$$

である．SS セカント条件では近似行列 B_k は正則化項を付加した行列 $\gamma_{k-1}(\nabla^2 g(x_k)+\nu_{k-1}I)$ の近似であるため，ν_{k-1} をうまく選ぶことで正定値性の保証が期待できる．修正 SS セカ

ント条件に基づく SR1 公式は通常の SR1 公式 (8)–(9) において，y_{k-1} を $\gamma_{k-1}z_{k-1}$ で置き換えることで得られるため，

$$B_k = B_{k-1} + \frac{(\gamma_{k-1}z_{k-1} - B_{k-1}s_{k-1})(\gamma_{k-1}z_{k-1} - B_{k-1}s_{k-1})^T}{s_{k-1}^T(\gamma_{k-1}z_{k-1} - B_{k-1}s_{k-1})}$$

$$H_k = H_{k-1} + \frac{(s_{k-1} - \gamma_{k-1}H_{k-1}z_{k-1})(s_{k-1} - \gamma_{k-1}H_{k-1}z_{k-1})^T}{\gamma_{k-1}z_{k-1}^T(s_{k-1} - \gamma_{k-1}H_{k-1}z_{k-1})}$$

で与えられる．したがって，修正 SS セカント条件に基づいたメモリーレス準ニュートン法の近似行列は，上記の更新公式で $B_{k-1} = I$ で置き換えて，

$$B_k = I + \frac{(\gamma_{k-1}z_{k-1} - s_{k-1})(\gamma_{k-1}z_{k-1} - s_{k-1})^T}{s_{k-1}^T(\gamma_{k-1}z_{k-1} - s_{k-1})} \tag{11}$$

$$H_k = I + \frac{(s_{k-1} - \gamma_{k-1}z_{k-1})(s_{k-1} - \gamma_{k-1}z_{k-1})^T}{\gamma_{k-1}z_{k-1}^T(s_{k-1} - \gamma_{k-1}z_{k-1})} \tag{12}$$

となる．次に，近似行列の正定値性を保証するようなパラメータの選択法を考える．Nakayama et al. [7] では，SS セカント条件に基づいたメモリーレス SR1 法の近似行列が正定値となる条件を与えており，それを今回の修正 SS セカント条件に基づく SR1 法で読み替えると，(11)–(12) が正定値となる条件は以下で与えられる．

命題 2. $\gamma_{k-1} > 0$ かつ $s_{k-1}^T z_{k-1} > 0$ であるとする．このとき, (11)–(12) で与えられる行列 B_k と H_k が正定値である必要十分条件は

$$\gamma_{k-1} \notin \left[\frac{s_{k-1}^T z_{k-1}}{z_{k-1}^T z_{k-1}}, \frac{s_{k-1}^T s_{k-1}}{s_{k-1}^T z_{k-1}}\right]$$

である． □

上記の命題の条件を満たすようなパラメータの選択法を考える．まず，$s_{k-1}^T z_{k-1} > 0$ となるような ν_{k-1} として，以下の選択法を採用する：

$$\nu_{k-1} = \begin{cases} 0, & \text{if } s_{k-1}^T y_{k-1} \geq \bar{\nu}\|s_{k-1}\|^2, \\ \bar{\nu}\left(1 - \frac{s_{k-1}^T y_{k-1}}{\|s_{k-1}\|^2}\right), & \text{otherwise.} \end{cases} \tag{13}$$

ここで，$\bar{\nu} \in (0, 1)$ はパラメータである．もし，$s_{k-1}^T y_{k-1} \geq \bar{\nu}\|s_{k-1}\|^2$ ならば $\nu_{k-1} = 0$ なので

$$s_{k-1}^T z_{k-1} = s_{k-1}^T(y_{k-1} + \nu_{k-1}s_{k-1}) \geq \bar{\nu}\|s_{k-1}\|^2.$$

が成立する．一方，$s_{k-1}^T y_{k-1} < \bar{\nu}\|s_{k-1}\|^2$ のときは，g が凸関数なので，$s_{k-1}^T y_{k-1} = (\nabla g(x_k) - \nabla g(x_{k-1}))^T(x_k - x_{k-1}) \geq 0$ であることに注意すると，

$$s_{k-1}^T z_{k-1} = (1 - \bar{\nu})s_{k-1}^T y_{k-1} + \bar{\nu}_{k-1}\|s_{k-1}\| \geq \bar{\nu}\|s_{k-1}\|^2$$

となり，どちらの場合でも

$$s_{k-1}^T z_{k-1} = \bar{\nu}\|s_{k-1}\|^2 > 0 \tag{14}$$

が成り立つ．次に，今回は γ_{k-1} として，

$$\gamma_{k-1} = \rho_{k-1}\frac{s_{k-1}^T z_{k-1}}{z_{k-1}^T z_{k-1}} \tag{15}$$

を選ぶこととする．ただし，ρ_{k-1} は，$0 < \rho_{\min} \leq \rho_{\max} < 1$ なる定数 $\rho_{\min}$，$\rho_{\max}$ に対して，$0 < \rho_{\min} \leq \rho_{k-1} \leq \rho_{\max} < 1$ を満たすパラメータである．このとき，(14) から $\gamma_{k-1} > 0$ なので，命題 2 の条件を満たすことは明らかである．

ここで，今回提案するアルゴリズムを述べる．アルゴリズムの大域的収束性を保障するために，Lee et al. [4] に倣い，Arimijo 条件を用いた直線探索を導入する．

アルゴリズム 1.

Step 0. 初期点 $x_0 \in \mathbb{R}^n$ と正定値対称な初期行列 B_0 と H_0 $(= B_0^{-1})$ を与える．パラメータ $\bar{\nu} \in (0,1)$, $0 < \rho_{\min} \leq \rho_{\max} < 1$, $\beta \in (0,1)$，および $\delta \in (0,1)$ を与え，$k := 0$ として Step 2．へ.

Step 1. パラメータ ν_{k-1} と γ_{k-1} を (13) と (15) によって計算し，B_k と H_k を (11) と (12) によって与える．

Step 2. 探索方向を以下によって計算する：

$$d_k = x_k^+ - x_k, \quad x_k^+ = \mathrm{Prox}_h^{B_k}\left(x_k - H_k \nabla g(x_k)\right). \tag{16}$$

Step 3. Armijo 条件：

$$f(x_k + \beta^i d_k) \leq f(x_k) + \delta\beta^i\left(\nabla g(x_k)^T d_k + h(x_k + d_k) - h(x_k)\right)$$

を満たす最小の非負数 i を見つけ，$\alpha_k = \beta^i$ とする.

Step 4. 更新式 $x_{k+1} = x_k + \alpha_k d_k$ によって，点列を更新する．

Step 5. $k := k+1$ として Step 1．へ.

通常，初期行列 B_0 と H_0 としては正定値対称な対角行列が選択される．上記アルゴリズムの Step 1. では，近似行列を計算しているように記述しているが，実際 d_k の計算 (16) においては，ベクトルの内積のみで計算可能であることを注意しておく．実際，$H_k\nabla g(x_k)$ は H_k の定義 (12) から，ベクトル積のみで計算可能であることが分かる．一方，$\mathrm{Prox}^{B_k}(z)$ の計算では，定理 1 において，$D = I$ として計算すればよいため，これも行列ベクトル積を用いていないことが分かる．したがって，提案法は問題の次元 n が大きな大規模問題に対しても適用可能である．

3　大域的収束性

この節では，前節で提案したアルゴリズムの大域的収束性を議論する．そのために，まず目的関数に対する仮定を設ける．

仮定 1. 関数 g は連続微分可能な凸関数であるとし，その勾配 ∇g はリプシッツ連続であるとする．また，関数 h は連続な凸関数であるとする．　□

まず，近似行列の性質について考える．

補題 3. 仮定 1 を満たしているとし，行列 B_k をアルゴリズム 1 で生成される行列であるとする．このとき，正の定数 m と M が存在して，すべての $k \geq 0$ に対して以下を満たす：

$$mI \preceq B_k \preceq MI.$$

ただし，対称行列 A と B に対して $A \preceq B$ は $B - A$ が半正定値であることを表す． □

上記の補題を用いることで，以下の命題を得る．

命題 4. 仮定 1 を満たしているとする．このとき，アルゴリズム 1 で生成される探索方向 d_k は以下を満たす：

$$\begin{aligned} f(x_k + \alpha d_k) \leq f(x_k) + \alpha \left(\nabla g(x_k)^T d_k + h(x_k + d_k) - h(x_k) \right) + O(\alpha^2), \\ \nabla g(x_k)^T d_k + h(x_k + d_k) - h(x_k) \leq -d_k B_k d_k. \end{aligned}$$

□

補題 5. 仮定 1 を満たしているとし，$\{x_k\}$ をアルゴリズム 1 によって生成される点列であるとする．このとき，x_k が問題 (1) の最適解である必要十分条件は $d_k = 0$ である． □

命題 4 より，アルゴリズム 1 の Step 3. の直線探索は実行可能であることがわかる．また，命題 5 から，終了判定条件として「$\|d_k\|$ が十分小さくなったら終了する」を考えることができる．さらに，上記の性質を用いることで，下記の大域的収束性の定理を得る．

定理 6. 仮定 1 を満たしているとし，$\{x_k\}$ をアルゴリズム 1 によって生成される点列であるとする．このとき，もし，目的関数 f が下に有界ならば，

$$\lim_{k \to \infty} \|d_k\| = 0$$

が成立する．さらに，最適解 x^* が存在し，点列 $\{x_k\}$ が有界ならば点列 $\{x_k\}$ は x^* に収束する． □

4 数値実験

この節では，提案したアルゴリズムの数値実験結果を報告する．テスト問題として，ℓ_1 正則化を用いたロジスティック回帰で生じる最適化問題：

$$\min_{x \in \mathbb{R}^n} \frac{1}{m} \sum_{i=1}^{m} \log \left(1 + \exp(-b_i x^T w_i) \right) + \lambda \|x\|_1$$

を使用し，$\lambda = 0.001$ とした．また，入力データ w_i，b_i $(i = 1, \ldots, m)$ として，UCI Machine Learning Repository [10] からデータセットとして “gisette”, “a9a”, “leukemia” の 3 つを選択した．初期点として，$x = (0, \ldots, 0)^T$ を用いた．

今回，我々は以下の方法の数値実験を行った．

FISTA : Nesterov の加速法を用いた近接勾配法 [1]
PNOPT : 準ニュートン型近接勾配法のソフトウェア [4,8]
mless-SR1 : 提案法 (アルゴリズム 1)

比較対象として，近接勾配法の代表的な方法である FISTA と準ニュートン型近接勾配法のソフトウェアである PNOPT を選択した．PNOPT は BSGS 更新公式に基づいて近似行列を生成しており，重み付き近接写像の計算には反復法を使用している．PNOPT はオプションで記憶制限 BFGS 法を用いることもできる．予備的な実験において，問題の次元 n が大きい場合に，通常の BFGS 法を用いた PNOPT は時間がかかりすぎたため，今回は記憶数を 10 とした記憶制限 BSGS 法を選択している．mless-SR1 ではパラメータを $\bar{\nu} = 0.01$, $\delta = 0.0001$, $\beta = 0.5$ とし，ρ_k を 0.1 から 0.9 まで 0.1 刻みで動かして実験した．また，終了判定条件は

$$\|d_k\|_\infty \leq 10^{-6}$$

とした．ここで，$\|\cdot\|_\infty$ は ℓ_∞ ノルムである．

表 1: 数値実験結果

データ名	gisette		a9a		leukemia	
データ数 m	1000		32561		38	
次元 n	5000		123		7129	
	反復回数	CPU time	反復回数	CPU time	反復回数	CPU time
FISTA	7025	1384.38	1123	21.28	1056	**4.27**
PNOPT	264	**137.62**	41	3.55	500	96.92
mless-SR1($\rho_k = 0.1$)	7743	1110.93	860	11.40	2148	11.84
mless-SR1($\rho_k = 0.2$)	8258	1147.18	383	4.81	3494	16.80
mless-SR1($\rho_k = 0.3$)	8563	1148.85	267	3.14	6678	32.87
mless-SR1($\rho_k = 0.4$)	9124	1213.54	227	2.62	11207	56.22
mless-SR1($\rho_k = 0.5$)	7003	932.56	182	2.16	12863	61.39
mless-SR1($\rho_k = 0.6$)	5307	696.41	189	2.19	15236	72.83
mless-SR1($\rho_k = 0.7$)	6676	871.12	164	1.91	16825	80.53
mless-SR1($\rho_k = 0.8$)	4869	630.99	170	1.99	17925	85.72
mless-SR1($\rho_k = 0.9$)	4271	548.00	151	**1.79**	17899	85.58

表 1 では，実験した方法の反復回数と実行時間をまとめている．まず，“gisette” に対する結果を見ると，反復回数，実行時間ともに PNOPT が優れていた．特に，反復回数に関しては比較した方法の中で群を抜いて少ない回数で終了している．次に優れていたのは mless-SR1($\rho_k = 0.9$) であった．両者の比較では，反復回数の面では PNOPT のほうが 20 倍近く少ないのに対し，実行時間では 4 倍程度の差となっている．これは，PNOPT では重み付き近接写像を計算するのに手間がかかっているためであると推測できる．次に，“a9a” についてみると，mless-SR1($\rho_k = 0.9$) の実行時間が最も少なかった．反復回数では PNOPT が優れているが，先ほどと同様の理由から実行時間では mless-SR1($\rho_k = 0.9$) が

勝っている．最後に “leukemia” の結果を見ると，実行時間の面で FISTA が最も優れており，mless-SR1($\rho_k = 0.1$) が続いている．

3 つの問題を解いた結果では，もっともすぐれた方法は三者三様となった．これは，次元数やデータ数の違いによるものとも考えることもできるが，どのような問題にどの方法が最も有効であるかはさらなる実験が必要だろう．

一方，mless-SR1 同士で比較すると，ρ_k の増減と方法の効率性には明らかな関係性があることが分かる．“gisette” と “a9a” の場合には ρ_k が大きくなるほど優れており，“leukemia” の場合は，ρ_k が小さいほうが優れていた．これもさらなる数値実験により，傾向を観察する必要があるだろう．

5 まとめと今後の課題

今回，重み付き近接写像を直接計算できるようなメモリーレス修正 SR1 法に基づいた準ニュートン型近接勾配法を提案した．この方法では近似行列は常に正定値であることが保証される．さらに，直線探索を導入したアルゴリズムに対する大域的収束性を議論した．数値実験では，既存の方法と比較して，提案法の有効性を検証している．数値実験結果では，問題ごとに三者三様の結果となったが，提案法がどのような問題に対して特に有効なのかをさらなる数値実験で明らかにしていく必要があるだろう．また，パラメータ ρ_k の選択法が数値的な効率性に大きく依存することも判明したため，その有効な選択法も今後の課題の一つである．

謝辞

本研究の一部は JSPS 科研費 JP17K00039，および，京都大学数理解析研究所の助成を受けて行われている．

参考文献

[1] A. Beck and M. Teboulle, A fast iterative shrinkage-thresholding algorithm for linear inverse problems, *SIAM Journal on Imaging Sciences*, 2 (2009), 183–202.

[2] S. Becker and M.J. Fadiliy, A quasi-Newton proximal splitting method, NIPS'12 Proceedings of the 25th International Conference on Neural Information Processing Systems, 2618–2626.

[3] W.Y. Cheng and D.H. Li, Spectral scaling BFGS method, *Journal of Optimization Theory and Applications*, **146** (2010), 305–319.

[4] J.D. Lee, Y. Sun and, M.A. Saunders, Proximal Newton-type methods for minimizing composite functions, *SIAM Journal on Optimization*, **24** (2014), 1420–1443.

[5] D.H. Li and M. Fukushima, A modified BFGS method and its global convergence in nonconvex minimization, *Journal of Computational and Applied Mathematics*, **129** (2001), 15–35.

[6] X. Liu, C.J. Hsieh, J.D. Lee and Y. Sun, An inexact subsampled proximal Newton-type method for large-scale machine learning, *arXiv preprint arXiv:1708.08552*, 2017.

[7] S. Nakayama, Y. Narushima and H. Yabe, A memoryless Symmetric rank-one method with sufficient descent property for unconstrained optimization, *Journal of Operations Research Society of Japan*, **61** (2018), 53–70.

[8] PNOPT website, `https://web.stanford.edu/group/SOL/software/pnopt/`, (最終アクセス日：2018 年 11 月 30 日).

[9] K. Scheinberg and X. Tang, Practical inexact proximal quasi-Newton method with global complexity analysis, *Mathematical Programming*, **160** (2016), 495–529.

[10] UCI Machine Learning Repository, `https://archive.ics.uci.edu/ml/datasets.html`, (最終アクセス日：2018 年 11 月 30 日).